

103.1033.01

IN THE CLAIMS:

Please amend the claims as follows:

Claims 1 to 11 (Cancelled)

¹
~~12.~~ (Previously Amended) A method of managing a file system for a file server, comprising the steps of:

receiving a file operation that signals a reservation operation for a file for which reservation has already been performed, said reservation operation specifying a new file size different from a current file size for the file;

comparing the current file size with the new file size;

in the case that the current file size exceeds the new file size, releasing any remaining block reservations for the file; and

in the case that the new file size exceeds the current file size, reserving for later allocation in the file system an additional number of unallocated blocks equal to a difference between a total number of direct and indirect blocks required by the new file size and a total number of direct and indirect blocks required by the current file size.

²
~~13.~~ (Original) A method as in claim ¹~~12~~, wherein the file system uses a write anywhere file system layout.

103.1033.01

3
14. (Original) A method as in claim 12, wherein the file operation that signals the reservation operation is a zero length write request.

4
15. (Original) A method as in claim 12, wherein the file operation that signals the reservation operation includes a parameter that specifies the file size.

5
16. (Original) A method as in claim 12, wherein the step of releasing remaining block reservations for the file further comprises the steps of:

resetting a flag in an inode for the file that indicates blocks have been reserved for the file; and

decrementing a reserved block count in a file system information block by a number of blocks still reserved for the file, the reserved block count indicating how many unallocated blocks have been reserved for files in the file system.

6
17. (Original) A method as in claim 12, further comprising the step of checking that a number of available blocks in the file system is greater than the additional number of unallocated blocks, wherein an error is returned in a case that the number of available blocks is less than the additional number of blocks.

103.1033.01

⁷
18. (Original) A method as in claim ⁶17, wherein the number of available blocks in the file system is determined by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks.

⁸
19. (Previously Amended) A method as in claim ¹12, further comprising the step of checking that the additional number of blocks does not exceed a remainder of a quota for an owner of the file, wherein an error is returned in a case that the additional number of blocks exceeds the remainder of the quota.

⁹
20. (Original) A method as in claim ¹12, further comprising the step of releasing reservation of blocks as those blocks are written to storage.

¹⁰
21. (Original) A method as in claim ⁹20, wherein the step of releasing reservation of blocks further comprises the step of decrementing a reserved block count in a file system information block by a number of released blocks, the reserved block count indicating how many blocks total have been reserved for files in the file server.

[
Claims 22 to 32 (Cancelled)

¹¹
33. (Previously Added) A file server that manages a file system, comprising:

103.1033.01

mass storage that stores files for the file system;

a processor that executes instructions to manage the file system; and

a memory storing information including instructions, the instructions executable by the processor to manage the file system, the instructions comprising: (a) receiving a file operation that signals a reservation operation for a file for which reservation has already been performed, said reservation operation specifying a new file size different from a current file size for the file; (b) comparing the current file size with the new file size; (c) in the case that the current file size exceeds the new file size, releasing any remaining block reservations for the file; and (d) in the case that the new file size exceeds the current file size, reserving for later allocation in the file system an additional number of unallocated blocks equal to a difference between a total number of direct and indirect blocks required by the new file size and a total number of direct and indirect blocks required by the current file size.

B2

¹²
34. (Previously Added) A file server as in claim ¹¹33, wherein the file system uses a write anywhere file system layout.

¹³
35. (Previously Added) A file server as in claim ¹¹33, wherein the file operation that signals the reservation operation is a zero length write request.

¹⁴
36. (Previously Added) A file server as in claim ¹¹33, wherein the file operation that signals the reservation operation includes a parameter that specifies the file size.

103.1033.01

15 11
37. (Previously Added) A file server as in claim 33, wherein the step of releasing remaining block reservations for the file further comprises the steps of:

resetting a flag in an inode for the file that indicates blocks have been reserved for the file; and

decrementing a reserved block count in a file system information block by a number of blocks still reserved for the file, the reserved block count indicating how many unallocated blocks have been reserved for files in the file system.

32 16 11
38. (Previously Added) A file server as in claim 33, wherein the instructions further comprise the step of checking that a number of available blocks in the file system is greater than the additional number of unallocated blocks, and wherein an error is returned in a case that the number of available blocks is less than the additional number of blocks.

17 16
39. (Previously Added) A file server as in claim 38, wherein the number of available blocks in the file system is determined by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks.

18 11
40. (Previously Added) A file server as in claim 33, wherein the instructions further comprise the step of checking that the additional number of blocks does not exceed a

103.1033.01

remainder of a quota for an owner of the file, and wherein an error is returned in a case that the additional number of blocks exceeds the remainder of the quota.

¹⁹
41. (Previously Added) A file server as in claim ¹¹33, wherein the instructions further comprise the step of releasing reservation of blocks as those blocks are written to storage.

²⁰
42. (Previously Added) A file server as in claim ¹⁹41, wherein the step of releasing reservation of blocks further comprises the step of decrementing a reserved block count in a file system information block by a number of released blocks, the reserved block count indicating how many blocks total have been reserved for files in the file server.

[Claims 43 to 53 (Cancelled)

21 ²¹ 54. (Previously Added) A memory storing information including instructions, the instructions executable by a processor to manage a file system, the instructions comprising the steps of:

²³ receiving a file operation that signals a reservation operation for a file for which reservation has already been performed, said reservation operation specifying a new file size different from a current file size for the file;

comparing the current file size with the new file size;

103.1033.01

in the case that the current file size exceeds the new file size, releasing any remaining block reservations for the file; and

in the case that the new file size exceeds the current file size, reserving for later allocation in the file system an additional number of unallocated blocks equal to a difference between a total number of direct and indirect blocks required by the new file size and a total number of direct and indirect blocks required by the current file size.

²²
~~55~~. (Previously Added) A memory as in claim ²¹~~54~~, wherein the file system uses a write anywhere file system layout.

B3 ²³
~~56~~. (Previously Added) A memory as in claim ²¹~~54~~, wherein the file operation that signals the reservation operation is a zero length write request.

²⁴
~~57~~. (Previously Added) A memory as in claim ²¹~~54~~, wherein the file operation that signals the reservation operation includes a parameter that specifies the file size.

²⁵
~~58~~. (Previously Added) A memory as in claim ²¹~~54~~, wherein the step of releasing remaining block reservations for the file further comprises the steps of:
resetting a flag in an inode for the file that indicates blocks have been reserved for the file; and

103.1033.01

decrementing a reserved block count in a file system information block by a number of blocks still reserved for the file, the reserved block count indicating how many unallocated blocks have been reserved for files in the file system.

²⁴
59. (Previously Added) A memory as in claim ²¹54, wherein the instructions further comprise the step of checking that a number of available blocks in the file system is greater than the additional number of unallocated blocks, and wherein an error is returned in a case that the number of available blocks is less than the additional number of blocks.

²⁷
60. (Previously Added) A memory as in claim ²⁴59, wherein the number of available blocks in the file system is determined by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks.

²⁹
61. (Previously Added) A memory as in claim ²¹54, wherein the instructions further comprise the step of checking that the additional number of blocks does not exceed a remainder of a quota for an owner of the file, and wherein an error is returned in a case that the additional number of blocks exceeds the remainder of the quota.

²⁹
62. (Previously Added) A memory as in claim ²¹54, wherein the instructions further comprise the step of releasing reservation of blocks as those blocks are written to storage.

103.1033.01

33
63. (Previously Added) A memory as in claim 62⁵¹, wherein the step of releasing reservation of blocks further comprises the step of decrementing a reserved block count in a file system information block by a number of released blocks, the reserved block count indicating how many blocks total have been reserved for files in the file server.
